RSTS-11 SIG
DECUS

DIGITGRAPH
P.O. Box 4249
Tucson, Arizona 85717

FEB 1976 RSTS-11 SPECIAL INTEREST GROUP NEWSLETTER VOL.3 NO.1

FROM THE EDITOR

OVER 100 RSTS USERS ATTENDED THE FALL DECUS SYMPOSIUM IN LOS ANGELES. THE MAIN OUTGROWTH OF THE MEETING WAS THE BRIDGE OF COMMUNICATION THAT WAS DEVELOPED BETWEEN THE SIG MEMBERS AND THE RSTS GROUP AT DEC HEADED BY ED CONVERSE, RSTS PRODUCT MANAGER. THE PEOPLE AT DEC WANT TO HEAR WHAT USERS WANT AND ASKED US AS A SIG TO LET THEM KNOW WHAT WE WANT. TO THIS END WE TRIED TO DEVELOP A "WISH LIST" OF USER NEEDS AND DESIRES FOR RSTS AT THE MEETING. THE FIRST COMPILATION OF THIS WISH LIST APPEARS IN THIS NEWSLETTER. IN ADDITION WE ARE PLANNING A SIG MEETING IN DALLAS MARCH 13TH AND 14TH(SEE THE ANOUNCEMENT IN THIS NEWLETTER) TO FURTHER DEVELOP THIS "WISH LIST" WITH THE RSTS PEOPLE FROM DEC.

PROBLEMS WITH "CUSP" PROGRAMS, PARTICULARLY "BACKUP", WAS ANOTHER MAJOR TOPIC OF DISCUSSION BETWEEN SIG MEMBERS AND DEC REPRESENTATIVES. ED HOPEY, RSTS/E SUPERVISOR, ASKED FOR USER NEEDS AND SUGGESTIONS FOR CUSP PROGRAMS AND INDICATED THAT THERE WILL BE INCREASED CUSP DEVELOPMENT.

I WOULD LIKE TO THANK EVERYONE FOR THEIR HELP IN GETTING THIS NEWSLETTER PUT TOGETHER. IN PARTICULAR THANKS TO STU KATZ FOR GETTING THE MEETING IN DALLAS PLANNED AND TO BILL SCONCE FOR THE BEAUTIFUL JOB HE HAS DONE AND THE MANY HOURS HE HAS SPENT GETTING THE "WISH LIST" STARTED.

THERE ARE 3 THINGS IN THIS NEWSLETTER FROM WHICH WE NEED YOUR RESPONSES: THE MARCH MEETING IN DALLAS; THE NEWLY DESIGNED RSTS USER PROFILE; AND THE "WISH LIST" QUESTIONAIRE. PLEASE FILL THEM OUT AND GET THEM TO THE PROPER PERSON AS SOON AS POSSIBLE.

BOB BRANTON
R. LA BAR & ASSOC.
2750 NORTHAVEN RD., SUITE 102
DALLAS, TX 75229
(214) 243-3451

THE RSTS SIG MEETING

THE MAJOR DEVELOPMENT OF THE SIG MEETING AT DECUS WAS THE FORMULATION OF WORKING GROUPS WITHIN THE SIG. IT IS HOPED THAT EACH MEMBER OF THE SIG WILL BECOME INVOLVED IN ONE OR MORE OF THESE GROUPS. BELOW IS A LIST OF THE SUB-GROUPS OF THE SIG AND THE NEW WORKING GROUPS AND THE PERSON HANDLING EACH.

A. SUB GROUPS:

COMMERCIAL RSTS USERS:

STU KATZ
COMMERCIAL COMPUTER SERVICES
401 FULLERTON PARKWAY
CHICAGO,ILL. 60614
(312) 929-7400

RSTS V4 USERS:

ROY SWATZELL JR.
V. A. HOSPITAL
ROOM 2 R 12
BIRMINGHAM, AL 35233
(205) 933-8103 EXT 239

EDUCATIONAL RSTS USERS:

HOWARD KIPPLE
NEW MEXICO JUNIOR COLLEGE
HOBBS,NEW MEXICO 88240
(505) 392-6526

B. WORKING GROUPS:

DECUS LIBRARY PROGRAM DEVELOPMENT:

BILL TABOR
FLORIDA COMPUTER, INC.
99 N. W. 183RD ST.
N. MIAMI, FLORIDA 33169
(305) 652-1710

PROGRAMMING STANDARDS DEVELOPMENT:

JACK BOUKNIGHT
PLYCOM SERVICES
P. O. BOX 160
PLYMOUTH, IN. 46563
(219) 936-2183

CUSP PROGRAM DEVELOPMENT:

HARVEY WIGGINS
CALLIER CENTER FOR COMMUNICATION DISORDERS
1966 INWOOD ROAD
DALLAS, TX 75235
(214) 638-1100

RSTS USER WISH LIST DEVELOPMENT:

BILL SCONCE
UNIVERSITY OF LOUISVILLE HSCC
P. O. BOX 1055
LOUISVILLE, KY 40201
(502) 588-5408

C. RSTS SIG SECRETARY(A BRAVE PERSON):

LINDA COHEN
FIRST NATIONAL CITY BANK
485 LEXINGTON AVE.
NEW YORK, NY 10017
(212) 559-6234

LETTERS

From: Robert Schwinger
Student Manager
Lawrence High
775 Bunker Rd
N. Woodmere, NY

Recently I sent a set of inquireies about RSTS V6A to Software Communication and received the follwoing responses (SPR #11-5917):

1. In response to the question of whether there is an easy way to patch CCLs in V06A-02, I was told that there is none presently, but that DEC is working on a way to add/delete/alter CCLs in the next release.

2. I asked whether some way existed to preserve the TEMPxx.TMP file in chaining via CCLs, so that programs to edit the current file without making a special copy of it were possible (e.g. a CCL command which would execute a program like the TEX REP feature on the DTSS.) The answer was that the structure of BASIC-PLUS and its use of the TEMPxx.TMP file prohibits this.

3. I asked about a method whereby only certain users (not necessarily in the same proj) could use a program not in their account. The response was that a good method would be to write a privileged (protection 232) program which would check his PPN before chaining to the program of his choice. Also it was indicated that BASIC-PLUS guards against wiping out any privileged, compiled program being run under a non-privileged account on CTRL/C, normal termination, or abnormal termination (e.g. an unexpected error.)

I have heard rumors that there is a full duplex modem made by VADIC which can be used over 1200 baud dial-up lines with RSTS/E.

I would appreciate any information anyone has on this topic and/or a Report Writer program.

Joan Salkin
Crown Zellerbach
1 Bush Street
San Francisco, CA 94119

415/823-5504

From: James Boettler
Director Computer Center
Talladege College
Talladege, Alabama

I've written some complaints and some suggestions to DEC. DEC is replied to me to fix the error in the DIRECT program, but I've heard nothing about the other problems. I'm enclosing the notes I sent to DEC in Maynard, since they may be useful to other RSTS users.

1. Problem with DIRECT program. Example: Type DIR(50,*) on our system (which has a system disk, a public disk, and a private disk). When program encounters an empty account, it prints out the file names of the programs found on the public disk for the previous account. Only solution for us is to specify a particular disk, like DIR DK0:(50,*),DK1:(50,*) then we will not get the spurious files printed out.

highest priority

2. Problem with TTYSET program and terminal characteristics function call (call 6, FIP 16 page 7-64 in RSTS/E Programming Manual 6A). We can set the terminal characteristic by either the program TTYSET or by the function call. But there presently is no way of probing the computer to find out the present characteristics. For example, for a particular program, we'd like to find the present characteristics on say up-arrow, change the up-appow to no-up arrow, execute the program, and then return the characteristics to the original values after the program is done. At present, I find no way of doing this. Suggested solution: modify the function so that there is data returned: in most cases the value returned should be 128 or 255 in each byte indicating present status of that byte.

3. Suggestion on the function INSTR. Right now you have it set so that program scans from left to right. Sometimes it is useful to scan from right to left instead. I suggest that you modify INSTR so that if the P% in INSTR(P%, A$, Y$) is negative, you start the scan at abs(P%) and scan to the left. Since no user is likely to use negative P%s in their programs, this should not harm any software already written. But in case someone does use negative P%s, perhaps this could be made an optional feature at sys-gen time, or be done by a patch.

4. Suggestion. For compatibility with other BASICs (like RT11, for example), I suggest you add the SEG function, so user can use either MID or SEG. I find myself simulating SEG most of the time, for I usually want to get segments from character n1 to n2; I find MID awkward in these cases.

5. LENGTH gives you core length of present program to next K. I'd like to have a program which tells me how many words left to next K.

The following patch to supply hard copy notation as to when things happen on a system was sent in by W. Jack Bouknight of Plycom Services:

Program Patch Notice

Program Name: INIT
Release #: 6A
Patch #: 1

Problem:

It would be nice to have INIT make notations as to when things happen during processing of .CTL files.

Solution:

This patch implements INIT commands TIME and DATE with a destination keyboard argument to print the current time and date out.

The syntax is the same as the FORCE command.

The resulting output will appear as follows:

```
TIME IS 12:3Ø AM          ^U
DATE IS 12-Dec-75         ^U
```

Procedure:

1. Take suitable steps to fetch the current source file for the program to be patched and place it in the workspace via the OLD command of RSTS/E.

2. Perform the following step(s) to properly modify the program.

```
11Ø     PRINT:
        PRINT "SYSTEM INITIALIZATION PROGRAM     VØ6A-Ø3.Ø1":
        PRINT:
        GO TO 13Ø
           IF F$ <> ""!INITIAL HERALD MESSAGE

3ØØ     GO TO 12ØØ
           IF LEFT(C$,4%) = "TIME"!PROCESS TIME PRINTOUT
```

```
31Ø     GO TO 13ØØ
           IF LEFT(C$,4%) = "DATE"!PROCESS DATE PRINTOUT

12ØØ    ON ERROR GO TO 97Ø:
        N$ = "TIME":
        NØ% = Ø%!FLAG TIME COMMAND

121Ø    E% = INSTR(1%,C$,":"):
        D$ = MID(C$,6%,(E%-5%))!STRIP ARGUMENT TEXT

122Ø    E% = Ø%:
        GO SUB 85Ø:
        GO TO 98Ø
           IF  E%
           AND E% <> -1%!CHECK FOR LEGAL KB #

123Ø    E% = M%(25%):
        M$ = N$ + " IS "!GET KB #, SET TEXT STRING

124Ø    IF NØ%
        THEN
           M$ = M$ + DATE$(Ø%)
        ELSE
           M$ = M$ + TIME$(Ø%)!ADD DATE OR TIME

125Ø    M$ = M$ + CHR$(9%) + CHR$(21%):
        GO SUB 69Ø:
        GO TO 2ØØ!ADD TAB, CTRL-U, FORCE TO KB

13ØØ    ON ERROR GO TO 97Ø:
        N$ = "DATE":
        NØ% = -1%!FLAG DATE COMMAND

131Ø    GO TO 121Ø!FINISH COMMAND PROCESSING
```

3. Take suitable steps to replace the program source(.BAS) file back on the proper backup medium and also place either the source or the compiled(.BAC) version of the program in the appropriate program library account.

December 15, 1975

Commercial Sub SIG

At our meeting in Los Angeles we had the opportunity to meet with all of our fellow users and to enjoy the numerous "bull sessions." For those members who have never attended a symposium, we can only indicate in brief terms the type of interplay taking place over the four day period.

At our sub SIG session we had the opportunity to hear from two Commercial RSTS users. The first was Mr. Ross Johnson of Johnson Printers in Downers Grove, Illinois, who gave his impression of RSTS as a new user, (delivery of his system was within the last month), and the second speaker was Murray Stinson of Scientific Placement, Houston, Texas, who is an established data processing professional, utilizing a well-established RSTS system.

Mr. Johnson indicated that his company had a bad experience approximately two years ago with a small manufacturer and suggested that future new users consult with a software house prior to chosing hardware, assuming that the software is not to be done "in-house. He also noted that as a new user he felt that a great service could be performed by the SIG, but he suggested that the new user sessions at the symposium be restricted to new users, thus limiting the sessions to proper topics.

Mr. Stinson discussed his experience with RSTS on a larger system, and as an experienced data processing person. He has both purchased and written his software depending upon what application needed. He described the problems in controlling a system that has "spread out like tentacles." Mr. Stinson also has the unique problem of having both the data processing responsibilities within the company and also the job of placing the data processing personnel with clients. He therefore has had to design his system to be as user independent as possible,while maintaining the necessary control over the users.

During our Sub SIG session we discussed very briefly the Commercial Sub SIG section of the newsletter is everybody's tool. We encouraged everyone to submit articles as well as announcements to the newsletter.

It was also determined at the symposium that a RSTS wish list should be developed and the RSTS SIG has begun the protocol necessary for such a list. In this regard the Commercial RSTS SIG has taken it upon themselves to host a Mid-Symposium meeting during the month of March in Texas. This session will be a two day weekend meeting open to RSTS users. Further information will be forthcoming.

Stu Katz
Commercial RSTS SIG Chairman

Commercial RSTS User Group Newsletter

It has been suggested that a helpful service our group could perform would be to conduct a survey of RSTS sites to obtain standard job descriptions and salary levels for the various positions of responsibility in each shop. This service would be particularly helpful to management of new and existing RSTS sites in establishing job descriptions and keeping salaries on par with the industry. The success of this endeavor will depend on adequate response from all geographic locations as well as detailed job descriptions, since both are important in comparing salaries. I plan to include a questionnaire for this survey in the next newsletter unless I get some negative feedback from group members. If there is enough response I will try to publish the results in the following newsletter.

Again, please send your problems, solutions, products, information, etc. to:

Murray Stinson
Scientific Placement, Inc.
2700 South Post Oak Road, Suite 2430
Houston, Texas 77027
713-621-9811

RSTS MARKETPLACE

DESCRIPTION	CONTACT
1.) GAAP - A General Accounting Application Package. A complete, interactive, on-line, multi-user system supporting the areas of general ledger, accounts receivable, accounts payable, and financial reports.	James Degnen Intelligent Systems Corp. 1 DeAngelo Drive Bedford, Mass. 01730 617-275-6576
2.) DBM - A general purpose Data Base Management Facility. User defined variable length records, variable length fields, and records with repeating groups. Acceptable characters definable for field values aids in the validation of input data. Direct, sequential, or indexed sequential access are featured.	
3.) General Utility Package - Includes programs for file formatting and listing, checking for possible file corruption, reviewing and modifying the contents of disk blocks, restructuring a data base for increased system efficiency, etc.	
4.) An on-line, interactive school administration package including advanced scheduling and grade/attendance reporting capabilities.	
5.) A commercial timesharing accounting system - Accumulates raw use data (connect hours, CPU time, storage) by system account number. (Such data automatically summarized in month-to-date reports thrice daily.) Applies rates to raw data for monthly billing. Provides for special and flat rates. This package may also be used for non-commercial installation to keep tabs on system utiliazation.	

RSTS Marketplace

DESCRIPTION	CONTACT
1.) A virtual sort subroutine for sorting up to 32,000 records for all files (include virtual) 2.) An easy to use line editor for editing program or ASCII data files. (automatic verification) 3.) A management information system for RSTS 4.) A jobshop scheduler for the allocation of labor and materials in manufacturing or maintenance 5.) A complete general ledger accounting system includes balance sheet, statements, journals and schedules such as accounts payable. 6.) An interactive payroll package with 941 & W-2. 7.) A complete automotive agency accounting system for GM, FORD and other dealers. An on-line system. 8.) A package of real estate programs - includes amortization schedules, data retrieval and others. 9.) Financial package for bidding and printing holdings of municipal and government bonds.	Al Cordes Houston Timesharing 9039 Katy Freeway #404 Houston, Texas 77024 713-461-5437
1.) Subroutines to phonetically code last names for storage and retrieval via standard phonetic techniques. 2.) Screen generator programs for the Tektronix 4023 buffered CRT terminal. 3.) Complete letter writing system using paragraph/letter files to build custom upper/lower case letters from form paragraphs. System allows insertions as well as a user composed paragraph..	Murray Stinson Scientific Placement, Inc. 2700 South Post Oak, #2430 Houston, Texas 77027 713-621-9811
A generalized Data Base and File Management System.	John Wright Florida Computer, Inc. 99 N. W. 183rd St. N. Miami, Florida 33169 305-652-1710
Multiple languages can be used with RSTS/E if you have OMSI's RT-11 emulator. OMSI-RT supports MCRO-11, OMSI APL, and other languages, in addition to DEC FORTRAN IV. 64K words of memory are recommended, but there are several 48K users.	Rusty Whitney OMSI Computing 4015 S.W. Canyon Rd. Portland, Oregon 97221 503-248-5903
1.) A consolidating Accounts Payable, General Ledger, Financial Statement package. 2.) A Payroll and Labor Distribution package. 3.) An Accounts Receivable and Sales Distribution package.	Jack Bouknight Plycom Services P. O. Box 160 Plymouth, Indiana 46563 219-936-2183

RSTS MARKETPLACE (cont)

1) "DATA BOSS" - A completely conversational generalized DBMS with multiple keys; immediate access on massive files; easily customized; contains its own special selection language with logical operators; forces documentation including a Generalized Applications Monitor; contains MULTI-LEVEL PASSWORD protection; allows for network structuring and linked data bases, and unlimited simultaneous users of the same Data Base; requires NO programming to use it; meets the vast majority of the CODASYL Data Base Task group language specifications.

2) COUNTRY CLUB PACKAGE - A complete system including accounts receivable, automatic dues, taxes and gratuities, locker control, cashflow analysis, member spend analysis, optional automatic gratuities calculation by waiter, daily managers report, food and beverage analysis, aging, member progress and activity report, all with complete audit trails.

3) FINSYS - A C.P.A. WRITE UP PACKAGE - Includes Transaction Registers, General Ledger, Multiple Format Profit and Loss Statements, Balance Sheets with comparisons to budget and last year, Employee Earnings Register, 941's, W-2's, Unemployment Reports, Source & Application of Funds. Consolidated and Departmental Statements, Accountants Worksheets.

4) PAYROLL - Includes hourly and automatic salary inputs, input listings & proofs, up to six miscellaneous deductions per employee, Federal, Multiple Local and State Taxes. Complete tips and meals calculations. Report and registers contain Current Pay, Period-To-Date, Quarter-To-Date and Year-To-Date information. Reports include Payroll Register, Check Register, Tax Liability Report, and Labor Distribution Reports, 941's and W-2's.

5) BILL OF MATERIALS EXPLOSION PACKAGE - With Production Planning and Inventory Control for manufacturers.

6) INVENTORY CONTROL SYSTEM - Including Sales Analysis, Reorder points, etc., for wholesalers.

John H. Wright
Florida Computer Inc.
99 N.W. 183 Street
North Miami, Fla 33169
(305) 652-1710

At the Fall Symposium. . . . . .

Ed Converse has taken Dick Angel's place as the RSTS Software Product Manager and has promised to submit a letter to this newsletter by Feb. 1, 1976 announcing DEC's policy regarding the distribution of RSTS V4 sources, listings, and the software support notes.

Additionally, Mr. Converse has a larger list of V4 sites than I published in the previous Resource Sharer and I am hopeful that we can combine our lists. This should be available for the next distribution.

GENERAL TECHNIQUES FOR MINIMIZING PROGRAM EXECUTION TIME IN RSTS-11 V4

Introduction

In production-oriented RSTS-11 systems, program execution time (and consequently multi-user system response time) can be optimized by following the general guidelines given below. Not each of these techniques, however, will necessarily be applicable at every installation; those techniques will have to be selected which best suit the particular configuration and the applications. Furthermore, although some of the techniques listed below simultaneously optimize both execution speed and program size, others optimize execution speed at the expense of demanding greater program size or disk space; in each case, therefore, the trade-offs involved should be carefully analyzed.

The guidelines given below are organized into four groups of descending importance: A, B, C, D. Those in group A are the most effective in minimizing program execution time; those in group D, the least effective. It is assumed that the system concerned has 28K of core, which should be prerequisite for any multi-user production-oriented system. The Record I/O option should also be considered prerequisite to any serious attempt to optimize response time.

A) Optimize programs' disk-accessing.

1) Optimize the Pack Cluster Size for each disk.

The pack cluster size is optimum when the entire SAT (Storage Allocation Table, stored as [Ø,1]SATT.SYS) can fit into the 256-word buffer in core beginning at SATBUF. To effect this, each RK disk should have a minimum pack cluster size of 2, and each RPØ2 disk should have a minimum pack cluster size of at least 8 and preferably of 16. RF disk units with less than 5 platters can have a pack cluster size of 1, but RF units with more than 4 platters should have a pack cluster size of 2.

2) Use private packs for production files rather than the public structure.

Whenever a file is opened on the public structure, the directories on every public disk have to be searched, either to verify the file's existence (OPEN FOR INPUT) or to ensure its non-existence (OPEN FOR OUTPUT). This overhead of searching more than one disk can be avoided by placing the file on a private pack.

3) Dedicate complete structures to large production files.

If possible, dedicate an entire private pack to a single large production file. This will ensure efficient directory organization and will minimize disk "seek" time for data accessing.

When it is necessary to put more than one file on the same pack, dedicate a whole account to each large production file. This will minimize directory-search overhead.

When it is necessary to put more than one file under the same account on the same disk, then create the large files before the small ones. This ensures better organization of the directory structures.

4) When possible, keep distinct files accessed by the same program on distinct disks.

If a program accesses more than one file and if all of these files are on the same disk pack, then disk head movement ("seeks") will be required every time the program's current reference to a file is different from its preceding file reference; thus a large percentage of execution time will be spent in moving the disk head back and forth. On the other hand, if each file referenced by the program exists on a distinct disk, then no head movement is required when changing from one file reference to another in the program; the heads will simply move once to their respective files and will change positions only when the location of the data itself requires it.

5) Pre-extend files to their maximum length.

Creating a file at execution time by sequential PRINTing of data, by sequential writing of a Virtual Core array, or by sequential PUTting of records entails a high overhead of repeated calls to non-resident parts of the File Processor in order to extend the file each time another sector is required. This needless overhead can be avoided if the file is already in existence at its maximum length or if it is created at its maximum anticipated length before any significant data is written upon it. Thus at the beginning of any program which is to create a long file, the highest record number should be PUT or the highest element of the Virtual Array should be written (even with meaningless data) before any other writing is done.

What has been stated here for data files applies also to scratch files. It is often best to create once and for all the maximum-length scratch file and thereafter to use it without ever deleting or re-creating it.

6) Use "Record I/O" disk accesses wherever possible.

Formatted ASCII I/O (INPUT, PRINT) is the slowest RSTS I/O. It examines each character one by one for delimiters, line terminators, etc. and executes conversions between internal binary forms and external ASCII representations for all numerical data.

Virtual Core I/O is fast for a given individual array, but it is not record-oriented. When a given logical record includes more than one item, Virtual Core I/O will scatter the elements of the same logical record into different areas of the disk, grouping them by their array type and name rather than by any relevance to the same record. This may multiply disk accesses in a program.

Record I/O (GET, PUT), as its name implies, has been designed for processing records, in either sequential or random-access mode. It packs data efficiently (e.g. numerical values are mapped in their binary form rather than converted to/from ASCII) and groups data by record organization. This minimizes the disk accesses needed for processing any record.

7) Optimize files' cluster sizes.

Since at any given moment the current retrieval information for a file is stored in core in the "window" of the file's FCB (File Control Block), retrieval overhead can be reduced to a minimum if the file's cluster size is such that the file does not need more than 7 total clusters. (The FCB has pointers to 7 clusters; if the file is longer than 7 clusters, then the FCB will have to be updated during processing by "turning the window".) Therefore, whenever possible, the file's CLUSTER-

SIZE should be set to a binary value (1, 2, 4, . . ., 256) equal to or greater than the maximum anticipated number of sectors in the file divided by 7.

Using large cluster sizes not only lessens "window turning" in the FCB but also lessens the need to read new sectors of directory information into FIBUF. Large cluster sizes, however, demand large amounts of free contiguous disk space.

8) Re-start from cleaned structures whenever possible.

When an account is first created and the first file is created under the new account, the linkages in the directory structures will be in the optimum order for fast and efficient directory handling. After a number of deletions, extensions, and new creations, however, this may no longer be true; directory linkages may then point forward and backward in whatever fashion was necessary at the time when the system was searching for free blockettes to update the directory. Consequently, to ensure that directory structures exist in a form optimized for fast access, it is best that disk pack periodically be re-structured. This can be done by using the DSKINT program to prepare a fresh pack and then transferring to that pack all files which are to be accessed on it.

Likewise an account should periodically be re-structured by first "zeroing" the account (under UTILTY or PIP) and then transferring to that account all files which are to be accessed under it. (This implies, of course, that existing files first be saved elsewhere before the account is "zeroed".)

9) Optimize record size if possible.

If a file has a cluster size larger than 1 and if the size of the logical records to be processed exceeds 512 bytes, then an explicit RECORDSIZE of 1Ø24 (or greater) will speed execution. This is true because RECORDSIZE defines the size of the user's buffer area and when the cluster size of the file allows it, the system will fill/empty the buffer in one disk access rather than in multiple accesses. If, however, the logical records to be processed are less than 512 bytes in length, then little is to be gained by departing from the default record size of 512 bytes.

10) Keep production accounts distinct from development accounts.

Because extensive file deletions, modifications, and creations can leave directory structures in an order ill suited for fast and efficient disk processing and because development work generally entails such file manipulations, development work should always be done under accounts distinct from those under which production files are kept.

B) Avoid run-time calls to the Garbage Collector.

The Garbage Collector is called at run time when the system's dynamic allocation of string space needs to reclaim deallocated string space. Calling the Garbage Collector allows RSTS high dynamic versatility but it also involves disk overhead, since the Garbage Collector is itself non-resident and does its work by using, when necessary, a scratch disk file. When processing speed and fast response time are important, programs should be structured so as to involve as little run-time garbage collection as possible. The following paragraphs delineate two general areas in which run-time garbage collection should be minimized.

1) Avoid OPENing files in the middle of a program.

Whenever a file is opened during program execution, buffer space in the user's program area is claimed. This is done most efficiently at the beginning of program execution when free contiguous space is more probably still available. If files are opened in the middle of program execution and if sufficient contiguous space is not available at that moment, then the Garbage Collector will be called to re-shuffle the available space. The overhead of garbage collection can therefore be largely avoided if all files which the program will use are OPENed as soon as possible. For the same reasons, re-opening a file should be avoided unless strictly necessary.

2) Avoid dynamic string assignments and manipulations as far as possible.

Consider four types of statements:

```
10 A$ = "ABCD" : B$ = SPACE$(5%)
20 C$ = A$
30 D$ = A$ + "E" :  E$ = LEFT(A$,2%)
40 LSET B$ = A$
```

Statements as in line 10 assign string variables and cause string space to be allocated in the user's area of core. Statements as in line 20 are re-definitions of existing string data; they do not cause new space to be allocated but simply point the left-hand variable to the existing string data to which the right-hand variable is already pointing. Statements as in line 30 are string-manipulations; they differ from the preceding type inasmuch as a new string is created, for which space must be allocated. Finally, statements of the type in line 40 are core-to-core data transfers; they do not cause any new space to be allocated but cause the string data to which the right-hand variable points to be copied into the string space (already allocated) to which the left-hand variable has been and is pointing.

Because statements such as those in lines 10 and 30 above must claim new string space and consequently may result in calls to the Garbage Collector, their use in a program can entail high run-time overhead. Consequently where execution speed is of concern, their use should be restricted to the minimum necessary. To this end the following guidelines will be helpful.

(a) Assign at the beginning of the program all string variables which will be used. Thereafter move string data, when possible, by the LSET and RSET verbs (which can null-fill or truncate) rather than manipulate the strings.

(b) Where string manipulations seem necessary, integer conversions (ASCII( ), CHANGE) and integer manipulations can often be substituted with less run-time overhead.

(c) Where actual string manipulations seem unavoidable, do not use intermediate variables or more than one statement if not necessary. For example, instead of

```
500 M$ = LEFT(Q$,3%)
510 N$ = RIGHT(P$,2%)
520 T$ = M$ + N$
```

a single statement such as

```
500 T$ = LEFT(Q$,3%) + RIGHT(P$,2%)
```

should be used.

CHANNEL 13

January, 1976

This portion of the Resource Sharer is offered in response to sentiments toward improved communication expressed by both users and DEC personnel at the Fall DECUS symposium. It is hoped that a two-way dialogue can be set up whereby DEC can consider user input in product planning and we (the users) can learn which features may be considered for implementation (or which are impractical!) and so be better equipped to plan for our own installations.

The following material represents the editor's interpretations of both sides, and due apology is offered for any inadvertent distortions. Furthermore, please note that DEC has made no commitment by offering us informal responses or comments. LA showed, however, that the RSTS people are interested enough in establishing communications to offer us time and effort in providing at least some of the answers.

Please by all means spend a moment with the questionnaire if any of its issues seem worth a comment. Also, please feel free to drop a line or to call to suggest new issues, air complaints, or whatever, since your input will form the basis for the next installment.

There have certainly been times in my personal "love affair" with RSTS when a trickle of information on things coming would have been welcome, to say nothing of an opportunity to air some thoughtful requests. Here, then, is a first effort toward providing both.

Bill Sconce, University of Louisville HSCC
P. O. Box 1055, Louisville, KY 40201
(502) 588-5408

****

CONTENTS FOLLOWING:

I RSTS/E Version 6B
II "New" Basic-Plus
III ¡Questions having Answers!
IV ¿Questions needing answers?
V Questionnaire

****

RSTS/E VERSION 6B

V06B will be out in the late fall of 1976, with a firmer date to be announced sometime in the second quarter. V06B will offer several features that were asked for in LA, and more important, DEC is still open for feedback in further refining the specifications. Drop me a line if you'd like to offer comments or suggestions.

Now, some features:

+ Sysgen procedure will no longer require DOS
+ It will be possible to disable non-existent or "down" devices
+ SEND-RECEIVE will handle longer (512 byte) message, with ability to use string variables and/or I/O buffers, and a caching facility for large messages
+ A controllable echo will permit "fill-in-the-blanks" format for terminals
+ Any newly-announced devices will be supported
+ There will be a CUSP for floppy disks
+ ISAM will be implemented in assembly language
+ An RT11 emulator will offer MACRO, ODT, LINK; DECtape and magtape support
+ Improved documentation
+ An improved patching facility; may even be semi-automated through keeping load maps on disk, etc.
+ Changeable CCL's
+ DECNET support for multi-CPU's: job-to-job communication, file sharing (via the DMC11 microprocessor-controlled communications interface)

---And, for Basic-Plus---

+ Statement lines of indefinite length
+ Comments imbedded within statements
+ Extended variable names
  - One alpha character plus from 0 to 29 alphanumerics or the character "."
  - %, $ suffixes will work as before
  - Some compile overhead, some .BAS storage overhead, but no .BAC or execution time overhead

"NEW" BASIC-PLUS

The Basic-Plus we all know and love is implemented interpretively, and is very powerful for debugging, since the guts of a program are available for dynamic inspection and even modification during a program run. B+ is even sometimes easy to write in, and has overall been a language worth expanding and upgrading.

There is, however, more in the works than just the extensions sketched above for V06B: DEC is working toward a "corporate standard" for Basic-Plus. Variously rumored as the "new Basic-Plus" or "Basic-Plus II", this would be a more conventional compiler type of language and would generate native object code. Resulting programs would be several times faster, and they'd be transportable to other operating systems, rather in the manner of current DEC FORTRAN. The new language will offer extended file processing (debocking, stored file attributes) through a new file processor interface. It will be possible to CALL subroutines.

We now come to the first questionnaire item, which concerns compatibility between the interpretive Basic-Plus (which will continue to be offered) and the "compiler" Basic-Plus. I've put it here instead of later because, in my opinion, the reasons it's been worth putting up with Basic-Plus' shortcomings are its functional conciseness and its rock-crushing debug power. Both of these translate to people productivity, and are part of the reason we are not large "brand X" shops in the first place.

¿O? The question is this: How important is it to us that the "new" Basic-Plus and the old not evolve in such a way that a program written to take advantage of, say, record deblocking or true subroutines, could no longer be debugged interactively?

¡1! *How about more I/O channels?*

RSTS' internal representation of channel number is a four-bit field. The resulting 16 channels are allocated as follows:

Monitor		Structure superimposed by the run time system (RTS) for Basic-Plus	
0	The jobs' console ("KB:" or "KB0:")		
1-14	available to the run time system (RTS)	1-12	user channels
		13,14	TEMPxx.TMP (B+source storage) channel for OLD' SAVE commands
15	Used for the "RUN" command		

More channel numbers will not be added: however, we might note that the most common case of bumping the ceiling (lots of terminals) was the motivation for creating the multi-TTY service facility.

¡2! *Files longer than 65535 records.*

This would require changes which would create incompatibilities with the existing file structure: therefore, NO. The same holds for clustersize greater than 256.

¡3! *Will RSTS/E for large systems (say a maximum 11/70) someday allow more than 63 users?*

No.

The resulting field-width problem would require a major re-work of the system tables.

Actually, it is not thought that many installations could exceed 63 from a practical performance standpoint. (But buy two /70's and network them!)

¡4! *Ability to lock out more than one block from access by other users.*

Will be available in V06B.

¡5! *A "PRINT-USING" type of facility for assignment statements, eg,*

```
LET   AS = B% USING "##,###"
```

This is one that DEC wanted, too: it has been checked out with some interest, but it isn't feasible. PRINT USING outputs character-by-character into a buffer, and it would be non-trivial to release the buffered result and turn it into a normal string. Too bad for us.

¡6! *An ability to use field statement on any string (not just buffers) would allow convenient and efficient dissection of blocked records, virtual arrays, etc.*

(FIELD statements are more efficient than the string functions LEFT, MID, etc., since the latter must create and load new strings and so eventually incur the overhead of garbage collection, particularly

in batching processing.

(It is for this reason that constructs of the form . . .

```
OPEN "######.TMP" AS FILE 12%:
KILL "######.TMP":
FIELD #12%, .....
```

although unofficial, have become a widely-used method of obtaining core space which can be redefined or subsetted.)

Unfortunately, we must continue on as we've been doing; the desired generalization of the FIELD statement is impossible due to internal structures.

¡7! *Why can't comments be included in data statements?*

Since the DATA statement syntax does not require quotes around strings, there would then be no way to code an "!" as part of the DATA.

¡8! *Could the upper limit on statement numbers be enlarged to 65535?*

Not reasonably on the PDP-11: getting around normal signed arithmatic would be pretty difficult.

¡9! *The "approximately equals (= =)" operator appears to behave peculiarly. For instance, given:*

```
A = .103 E -10
B = 0.
```

then (A = = B) *is "FALSE".*

Indeed, this is confusing if one is expecting to use "= =" to test for arithmatic approximation or rounding. The test actually concerns significant figures, or the value as printed:

"expl = = exp2" is exactly equivalent to
"NUM$(expl) = NUM$(exp2)"

This construct was actually intended for the CAI environment, to avoid having a program tell a naive user "Your answer of 2 is wrong; 2 is correct" after testing his response against 2.00000003+, which prints as 2.

This is properly documented in section 2.5.4 of the Basic-Plus manual.

¡10! *$BACKUP is reported to be in serious trouble: it's much too slow, won't handle ANSI labels, doesn't copy open virtual arrays, etc., etc.*

DEC's RSTS people don't like to talk about BACKUP these days, the reaction has been so terrible. Seriously, there seems to be no argument that something must be done if RSTS is to have a functional capability for on-line backup.

It should perhaps be said that the original program (from which the current $BACKUP was developed) was written in the field, and not as part of the formal RSTS project. A new BACKUP is forthcoming, however: it will probably use interactive dialogue instead of a command line, and have the speed of its directory operations improved à la $DIRECT/

DEC has promised that the new program will be available with V06B, although every effort will be made to provide it as soon as possible.

There also exists at least one improved version available from

a RSTS SIG member for a (largish) nominal fee: Michael Crowley, Alden Press, 2000 Arthur Avenue, Elk Grove, IL 60007, (312) 640-6000.

¡11! *Improve system checkout performed by INIT code (bootstrap or START time).*
Boot checks will be much expanded in V06B, and will include such things as automatic lockout of non-existent devices (no more crashes by reference to a missing line printer, etc.).

¡12! *Extended PIP is slow!*
Yes; one can save time by using the old, standard PIP where the expanded functions of PIPEXT (such as wildcards or DOS file handling) are not required.

For V06B, some effort will be made to bridge the gap between capability and speed, possibly by a program which CHAINS.

¡13! *RSTS $EDIT needs a new manual.*
EDIT under RSTS will have its own manual, spearate from DOS/BATCH, beginning with V06B.

¡14! *In the future, could there be more/improved documentation of how RSTS/E works internally?*
Definitely. There is presently in the works a "system programmers manual", written in assembly-language terms. It's to have diagrams of system tables, description of system functions, run time system interfaces, scheduling algorithm, and so on. It may be worth noting that publication of such data will itself tend to stabilize the specifications.

¡15! *Is there planned support for graphics devices under RSTS/E (plotters, CRT's, etc.)?*
Not in the sense of PLOT subroutines, or special device drivers, at monitor level. However, a growing family of devices interfacing to RS232 offer graphics capability. These will obviously work as RSTS/E terminals, and are often controlled by escape sequences, which are supported. Basic-Plus routines for this kind of work should very likely be contributed to the DECUS library.

¡16! *Is it true that RSTS/E is going to support RT-11 compilers and utilities?*
Yes, in version 06B. It'll be a separate-cost item.

¡17! *Will there be a version of Basic-Plus (16k user) available on smaller PDP-11's than at present?*
No. However, the BASIC-11 language (which is available on smaller machines, will evolve toward being a strict subset of the eventual "corporate standard" BASIC/ Since Basic-Plus (as well as the forthcoming RSTS "compiler Basic-Plus") will also evolve toward the standard, it's likely that all dialects will get closer together.

¡18! *Will FILES-11 support become available under RSTS/E?*
Record structuring support (logical records, blocking/deblocking, etc.) will become available with V06B. However, out of consideration for compatibility with existing software, no change toward FILES-11 file structure is contemplated.

¡19! *Multiple-CPU support for RSTS/E?*
Yes; provided by DECNET, the first components of which will be released with V06B.

¡20! *In UPDATE mode: the ability to extend a file; the ability to specify a RECORDSIZE.*
No and yes. UPDATE extension is not practical since file-windowing tables, "dirty-data" switches, etc. are kept only in core, and would require complicated updating with file extension.
UPDATE RECORDSIZE will be available in V06B.

¡21! *Couldn't we have unified cursor control for the various scopes?*
Everyone admits that incompatibility between the VT05 and the VT50 was an unfortunate event of history. DEC assures us that further offerings in the VT50... series will have a unified cursor addressing scheme.
As far as tubes in general are concerned, it's obvious that the number of configurations of different scopes an installation could have is virtually limitless, and there would be no way for one packaged operating system to conveniently support them all. On the other hand, if the need justifies a live-or-die-by-one's-own-hand approach, the terminal service routines are (and will continue to be) supplied in source form. (Be sensitive to your situation; this comment was <u>not</u> suggested by DEC!).

¡22! *Literals should be assigned storage space only once even though used several times in a program.*
True, they should; but there would be structure problems in the current Basic-Plus, so it's not likely.
Handling this kind of thing is normally expected of compiler-type processors, however, so it's reasonable to expect that the forthcoming "compiler Basic-Plus" will be able to handle it efficiently.

¡23! *There should be a way to write a pure comment line that does not use up object-time core.*
They use core because we sometimes GOTO or GOSUB them, and creating some special designation for a "pure" comment would change the syntax of the current language in a direction not consonant with the eventual standards.
On the other hand, the V06B Basic-Plus syntax extensions to statement length and interspersed comments will go a long way toward circumventing the need.

¡24! *We'd like to see RSTS support for the TECO editor.*
It's on the way (DEC people like TECO, too), but will probably be offered through the DECUS library instead of as a standard part of RSTS/E.

¡25! *Virtual arrays with a recordsize greater than 512.*
This limitation is tied up in the nature of Basic-Plus' subscripting algorithms, and they're reasonably complex; expansion isn't feasible.

The following questions are the ones for which DEC would like users' input. Answer any/all you like; add any pertinent comments; and finally, include any suggested questions you'd like to see in the next CHANNEL 13. Your answers don't have to be formal, nor do they even have to appear on the questionnaire form. Xeroxes of your own scribbled notes are fine; anything goes.

Questions ¿0? to ¿10? are provided with boxes which may be checked on the questionnaire form. These were arranged together in the hopes of saving everyone time. The remaining questions are really just suggested departure points for any thoughts or comments you may have; here again, please feel free to offer informal notes, a phone call, or whatever.

¿0? SEE: "NEW BASIC-PLUS" ABOVE.

¿1? *File access statistics need a "date modified" to, among other things, trigger a timely backup.*
There isn't room in the current UFD allocation, so something would have to go. Would we like to have a sysgen option for "date modified" to be kept instead of "date created" or "time created"?

¿2? *Multiple RP swap files; for instance SWAP0 on DP0:; SWAP1 on DP1:.*
This one's on DEC's "wish list", but not yet into the functional specs. How important is it to us?

¿3? *Multiple statements in an IF, like:* IF A% THEN xxxx: xxxx: xxxx:
ELSE xxxx: xxxx: xxxx
The current syntax of Basic-Plus won't allow such constructions. Would we want the "new" B+ to provide for them?

¿4? *Basic-Plus facilities for:*
a) Unsigned integers?
b) Double-precision integers?
c) Selectable-precision reals?
d) ABS%, SGN% functions to give integer results?
3) MODULO function?
f) Built-in values TRUE (-1%), FALSE (0%)?
g) Inverse function for DATE$(n)?
h) Opening a file read-only?
i) An object-time TRACE?
j) CVT%S, CVTS%, etc. for SORT-format strings

¿5? *Some way to modify one line of a multi-line B+ statement.*
This one could be fairly complex (DEC looked into it back around V05A), and may have no easy answer. On the other hand, it's needed, especially with statements of indefinite length coming. How badly?

¿6? *Basic-Plus would benefit from built-in table-handling facilities.*
One possible implementation would be functions to maintain sequential-table arrays and to do lookups - that is, to add an entry, delete an entry, and to do a binary search.

¿7? *How about an ability to designate a terminal for receive-only service, like a remote line printer?*
This would get around the overhead associated with the TTY handler, etc. It could be done either as a sysgen option (one-time) or a particular terminal could be made switchable between keyboard and receive-only status. Presently under consideration for V06B.

¿8? *Ability to assign one buffer tc more than one channel would improve some processing, such as simple updates, PIP, BACKUP.*
This has been done in at least one instance by a custom software house. How important would it be to us?

¿9? *An alternate work area or alternate swap slot to permit temporary, non-destructive interruption of one job by another, eg,*
ROLLOUT (save current job)
RUN $PIP (run something else)
ROLLIN (restore job & continue)
It would undoubtedly be beautiful, but might be unexpectedly difficult to implement. How important?

¿10? *A special "reconstruction" program to recover whatever is saveable from a really corrupted disk pack.*
Worth its weight in gold although not used very often. Could be run stand-alone and booted like ROLLIN. If necessary it could delete all non-recoverable files in the hopes of salvaging the rest.

****

¿11? *The system accounting routines could benefit by some refinement...*
Split device usage; ie, make LP: MTx: PK: KB: separate. Provide a user accounting area option where user can DUMP, or where LOGIN and LOGOUT can record data (slots in area should be created and deleted by the ADD and DELETE PPN sys calls, although area itself could reside in a file). Provide accounting of lines/pages printed. Don't bother with $ACCT.SYS in $REACT; provide an as-required DUMP from MFD and user accounting area instead.
Do you have comments/suggestions?

¿12? *Queue Manager...*
Que file should not be corrupted by a system crash. Queuing for, say, "03:00 tomorrow" should be possible and should not result in immediate processing because it's later than 03:00 today. It should be possible to queue to anyone of a class of line printers.
Do you have comments/suggestions?

¿13? *Spoolers: similarly here; do you have suggestions for improvement?*

¿14? *Auto-dial support and support for 1200 baud half duplex.*
These are under consideration at DEC but could be implemented in any of a large variety of ways. They would appreciate user input.

QUESTIONNAIRE

¿0? How important is it that the various Basic-Plus'es evolve compatibly?
☐ Vital ☐ Important ☐ Merely Nice ☐ Indifferent

¿1? Sysgen option for "date modified" to replace
☐ "date created" ☐ "time created" ☐ forget it

¿2? Multiple RP swap files?
☐ Vital ☐ Important ☐ Merely Nice ☐ Indifferent

¿3? Multiple-statement IF's?
☐ Vital ☐ Important ☐ Merely Nice ☐ Indifferent

¿4? Several new facilities...

a) Unsigned Int's.....	☐ Vital	☐ Important	☐ Merely Nice	☐ Indifferent
b) Dbl-prec Int's.....	☐ Vital	☐ Important	☐ Merely Nice	☐ Indifferent
c) Selct-prec reals....	☐ Vital	☐ Important	☐ Merely Nice	☐ Indifferent
d) ABS%, SGN%.....	☐ Vital	☐ Important	☐ Merely Nice	☐ Indifferent
e) MODULO.....	☐ Vital	☐ Important	☐ Merely Nice	☐ Indifferent
f) TRUE, FALSE.....	☐ Vital	☐ Important	☐ Merely Nice	☐ Indifferent
g) DATE$ inverse.....	☐ Vital	☐ Important	☐ Merely Nice	☐ Indifferent
h) Open read-only.....	☐ Vital	☐ Important	☐ Merely Nice	☐ Indifferent
i) TRACE.....	☐ Vital	☐ Important	☐ Merely Nice	☐ Indifferent
j) CVT%S etc.....	☐ Vital	☐ Important	☐ Merely Nice	☐ Indifferent

¿5? Editing within multi-line B+ statements?
☐ Vital ☐ Important ☐ Merely Nice ☐ Indifferent

¿6? Table Handling Functions?
☐ Vital ☐ Important ☐ Merely Nice ☐ Indifferent

¿7? Receive-only KB Facility?
☐ Vital ☐ Important ☐ Merely Nice ☐ Indifferent

¿8? Multiple buffer assignment?
☐ Vital ☐ Important ☐ Merely Nice ☐ Indifferent

¿9? ROLLOUT-ROLLIN Lookaside?
☐ Vital ☐ Important ☐ Merely Nice ☐ Indifferent

¿10? "Reconstructor" Program?
☐ Vital ☐ Important ☐ Merely Nice ☐ Indifferent

Bill Sconce
University of Louisville HSCC
P. O. Box 1055
Louisville, KY 40201

(502) 588-5408

This new RSTS user profile was developed by Elizabeth Matlack of Dow Chenical and Howard Kipple of New Mexico Junior College. Please send your completed profile to:

Linda Cohen
First National City Bank
485 Lexington Ave.
New York, NY 10017

RSTS USER PROFILE

Date of Installation: _____

Contact Person _____

Address: _____

City, State, Zip _____

Company _____ Phone _____

For multiple installations please reporduce this form and respond for each installation.

Respond with quantity of each item.

Processor: 11/20 _____ 11/40 _____ 11/45 _____ 11/50 _____ 11/70 _____ Other _____

Operating System: V4A ____ V5A _____ V5B _____ V5C _____ V6a _____ V6B _____ Other _____

Languages: COBOL _____ Basic + _____ Fortran _____ Dibol _____ Other _____

Memory Size: MOS _____ Cache _____ Core _____ Bipole _____ Other _____

Vendor of other _____

RJE 2780 Corrected to: 360 ____ 370 _____ Other _____

Card Reader: CR11 _____ CM _____ Other _____

Card Punch: _____

Line Printer: LS 11 _____ LP 11 _____ Other _____

Terminals: TTY _____ LA 30 _____ LA 36 _____ Other _____

Vendor _____.

CRTS: VT 05 _____ VT 50 _____ VT 52 _____ Other _____ Vendor _____

Lines: DH 11 _____ DN 11 _____ DL 11/A _____ DL 11/E _____ Other _____

Disk Drives: RK 02 _____ RK 05 _____ RP 02 _____ RP 03 _____ RP 04 _____

RS _____ RF _____ Other _____ Vendor _____

Mag Tape: 7 Track _____ TU 10 _____ TU 16 _____ 9 Track _____

Other _____ Vendor _____

DECTAPE: TU 56 _____

Floppy Disk _____

Current Applications: _____

_____

Need Software For: _____

_____

_____

MARCH SIG MEETING IN DALLAS

RESERVATION SLIP

NAME _____

ADDRESS _____

CITY _____ *STATE* _____ *ZIP* _____

PHONE _____

BUSINESS ADDRESS _____

BUSINESS PHONE _____

DALLAS MARRIOTT

*$20-$30** *SINGLE* ( )

*$25-$38** *DOUBLE* ( )

ARRIVING DATE _____

DEPARTING DATE _____

PLEASE INCLUDE ONE NIGHT'S DEPOSIT WITH THIS SLIP---

RETURN BEFORE JANUARY 26, 1976

**IF WE HAVE ENOUGH PEOPLE, THESE RATES WILL BE REDUCED TO GROUP RATES.*

MARCH SIG MEETING IN DALLAS

January 6, 1976

As promised at the Fall DECUS Symposium, we are holding a Mid-Symposium meeting of the RSTS-SIG. This meeting will be held in Dallas, Texas at the Dallas Marriott on March 13th and 14th. On the agenda for this meeting will be the preparation of a detailed "wishlist" for the coming symposium in Atlanta. We will also schedule time for discussion of certain CUSP problems as an extension of the CUSP meeting in Los Angeles. The remainder of the time will be opened for discussion on whatwever topics are deemed important. The meetings will be from 9:00 AM to 5:00 PM on Saturday and from 9:00 AM to 1:00 PM on Sunday. Informal discussions will take place throughout the entire weekend.

The meetings were scheduled for the weekend by the request of a number of the members. The Marriott was chosen for its pools, tennis, food, close proximity to the Dallas area, transportation and meeting rooms. We are hoping to take benefit of the discount rates, depending on the number of people attending.

Please fill in the enclosed reservation slip and return it to Stu Katz as soon as possible so that we can make arrangements no later than January 26, 1976, in order to take advantage of the group rates. Your attendance will insure a good "wish list." We have a committment from the RSTS Product Panel to have DEC personnel present at this meeting if enough RSTS users are present. This is our chance to expand the usefulness of the SUG and our semi-annual symposium.

If you have questions, please contact---

Stuart Katz
Commercial Computer Services
411 Fullerton Parkway
Chicago, Illinois 60614
(312) 929-7400

Murray Stinson
Scientific Placement, Inc.
2700 Post Oak Road
Houston, Texas 77027
(713) 621-9811

Bob Branton
R. Labar & Assoc.
2750 Northhaven Road
Dallas, Texas 75229
(214) 243-3451

RSTS - SIG